STA 32 Program 3, due Friday, Feb 21

Problem 1

In this problem we will verify that Chebyshev's inequality holds. Use the command `X = rnorm(1000)` to draw a sample from a normal distribution. Then, perform the following tasks:

(a) Calculate and report the mean and standard deviation of `X`.

(b) Calculate the different between each data and the mean, then calculate the number of items that are less than $-k\sigma$, or larger than $k\sigma$, when $k = 1, 2, 3$. You may do this with the command
`sum(X - xbar < -k*sigma | X - xbar > k*sigma)`
Using this, calculate and report $P(|X - \mu_X| \geq k * \sigma_X)$.

(c) Check and report if the probability found in part (b) is less than $1/k^2$ for each value of k.

Use a new set of X values for each k. Report the mean, standard deviation, and the probability. Round all quantities to 3 decimal places.
You should build a function that takes in a vector X, a value k and outputs the required quantities for a given k. A sample outline is below (you do not have to use this format).

```
MyChebyShev = function(X,k){
...
return(c(xbar,sigma,probability,check))
}
set.seed(1001)
X1 = rnorm(1000)
MyChebyShev(X1,1)
X2 = rnorm(1000)
MyChebyShev(X2,2)
X3 = rnorm(1000)
MyChebyShev(X3,3)
```

Problem 2:

In the 2008 election, there were 131,257,328 voters. 69,456,897 voted for Obama. Let p be the probability that a voter voted for Obama.

(a) Find and report p.

(b) If we select 10 voters at random, what is the probability that 7 voted for Obama? Justify why you can use the Binomial distribution, and then use it to calculate $P(X = 7)$.

(c) Create a function that takes in as its arguments `n` and `N` and `p`, that creates N random samples of n voters using the function `rbinom`, and calculates the mean ($\hat{p}$) for every random sample. Using your function, report the mean for a single sample of size 10 ($N = 1, n = 10$). Set your seed to 9999 before running your function.
Remember, `rbinom` will take many samples of a particular size, and report back the number of successes per sample (see handout 5 for an example) - use this to find the mean $\hat{p}$.

```
set.seed(9999)
yourfunction(N = 1, n = 10, p = p) #The second p
 here is the answer you found in part a
```

(d) Using $N = 1000$ (equivalent to 1000 possible polling samples of size n), and $n = 10, 100, 1000, 10000$, find the mean of all the values of $\hat{p}$, and the variance of all the the $\hat{p}$ (i.e. the variance of the mean). What do you observe as your sample size n grows? (Again, set your seed to 9999 again before doing this problem)

```
set.seed(9999)
yourfunction(N = 1000, n = 10, p = p) #and different n values
#You will get 1000 values for each n. Find \hat{p} for each value then
 take the mean of the 1000 values.
```

Problem 3:

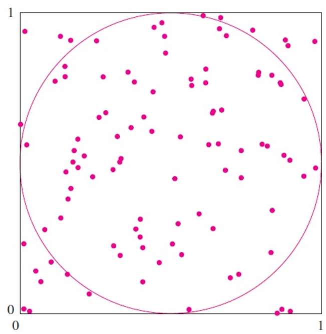
The goal of this problem is to implement a non parametric bootstrapping algorithm, estimate the standard deviation of the mean AND median.
Write a function which takes in a vector `X`, and the number of bootstrap samples, `B`. It should output the estimated mean, median, standard deviation of the mean, and standard deviation of the median for a given `X`.

(a) Using the files `Dataset1.txt`, `Dataset2.txt`, and `Dataset3.txt`, use your program to sample $B = 100, 1000, 10000$ bootstrap samples, and report back the bootstrap estimates for the mean, median, standard deviation for the mean, and standard deviation for the median. Set your seed to **1001** (so for each B and each dataset, you should return 4 values. Thus in total, you should have 36 values).

(b) Plot histograms for `Dataset1.txt` and `Dataset2.txt` (and show them), and use the function `unique` to find the unique values they take on. Do they look as if they come from a particular distribution? If yes, what distribution do you think they could be from? (you don't need to report the parameter values)

Problem 4:

The following figure suggests how to estimate the value of π with a simulation. In the figure, a circle with area equal to $\pi/4$ is inscribed in a square whose area is equal to 1. One hundred points have been randomly chosen from within the square. The probability that a point is inside the circle is equal to the fraction of the area of the square that is taken up by the circle, which is $\pi/4$. We can therefore estimate the value of $\pi/4$ by counting the number of points inside the circle, which is 79, and dividing by the total number of points, which is 100, to obtain the estimate $\pi/4 \approx 0.79$. From this we conclude that $\pi \approx 4(0.79) = 3.16$. This exercise presents a simulation experiment that is designed to estimate the value of by generating 1000 points in the unit square.



The following steps outline how this can be done:

1. Generate 1000 x coordinates $X_1^*, ..., X_{1000}^*$. Use the uniform distribution with minimum value 0 and maximum value 1. (To generate values from a uniform distribution, you can use the function `runif()`)

2. Generate 1000 y coordinates $Y_1^*, ..., Y_{1000}^*$. Use the uniform distribution with minimum value 0 and maximum value 1.

3. Each point (X_i^*, Y_i^*) is inside the circle if its distance from the center $(0.5, 0.5)$ is less than 0.5. For each pair (X_i^*, Y_i^*), determine whether its distance from the center is less than 0.5.

4. Count the number of points that are inside the circle and estimate π.

Set your seed to **9999**. Do this experiment using $n = 1000, 10000$ and 100000 points and report your estimated π value. Round your estimations to four decimal places.

How to turn in homework

I am not going to set up templates this time. Just like the first programming assignment, write/type up your answers, and print out a hard copy of the code for turning in. Once again I DO NOT WANT TO SEE ANSWERS AND CODES MIXED TOGETHER. Then email us your code.